# FilterTree: a Repeatable Branching XES Editor

Sander J.J. Leemans
RWTH, Aachen, Germany
s.leemans@bpm.rwth-aachen.de

***Abstract*—A large fraction of process mining efforts is spent on event data preparation: the step between data extraction and the subsequent analysis using process mining tools. Data preparation may be repetitive and is typically performed in a trial-and-error way. In this paper, we introduce the FilterTree XES and CSV editing tool, which allows for the programmatic chaining of XES and CSV filters, allowing for repeatable event data preparation. The FilterTree tool is platform-independent and open source.**

***Index Terms*—process mining, event log filtering, XES editor**

## I. Introduction

Common folklore in process mining tells that of the time spent on process mining projects, 80% is spent on preparing event data, and only 20% is spent on analysis. Massaging event data into an event log is a trial-and-error process, which may involve selecting activity columns, selecting case columns, altering data types, combining columns, parsing timestamps, filtering, addressing data quality issues, selecting activities, computing aggregate columns, etc. The repetitiveness of this process is captured by several process mining methodologies, such as [5]: even in the final phases of analysis, the data preparation may have to change, for instance after discovery of data quality issues or after analysis questions have been adjusted, thus requiring a slightly different view on the event data.

In this paper, we propose a tool to import CSV or XES files, and edit XES event logs by means of filters. A *filter* reads an XES (or CSV) file from disk and writes an adjusted XES file to disk. Filters are organised by the user into a *filter tree*, which specifies the filters with their parameters. In a filter tree, most filters are sequential, that is, one is applied to the result of its predecessor. It is also possible branch the filter chain, where a single filter may have more than one subsequent filters.

## II. Significance, Innovations & Main Features

Every process mining project starts with extracting event data. Except in straightforward "projects" where public data is used in a standard process mining technique, event data preparation is a necessary next step before the actual process analysis can commence. Many existing tools can perform event data preparation and apply filters to an event log, and a few tools can perform edit operations on XES logs [1], [3], [4], [6]. To the best of our knowledge, there is no tool that is a combination of:

- XES-based, with support for CSV. Importing CSV files is critical, but advanced process mining operations require XES concepts, such as trace attributes, log attributes, summing trace outcomes, etc.
- Repeatable. The same set of filters can be applied to a new event log, thereby repeating the analysis without manual filtering steps, or to slightly change how an event log is prepared, requires easy repeatability.
- Branchable. In process mining projects, several different perspectives may be necessary to answer the analysis questions. These perspectives may require different event logs. Branching allows to re-use parts of chains of filters.
- Disk based and file-manager friendly. A common problem encountered is CSV or XES files that are too big to visualise or load into process mining tools. While filtering, logs should be handled disk-to-disk as to support any log that fits on disk.
- Offline. An offline tool provides privacy and confidentiality, and does not need to upload datasets.
- Extensible. There is always another, unsupported, filter, thus the tools needs to be easily extensible.

The FilterTree tool aims to satisfy all of these properties.

### A. Notable Plug-ins

CSV files can be either row-based or column-based; for both, FilterTree has a plug-in. In a row-based CSV file, each row represents an event. Using the filter `CSV to XES`, the only parameter necessary is the name(s) of the column(s) of the CSV file that contain the trace identifier, that is, the column that tells us which case the event belongs to. This column or combination of columns is copied to the trace level as its `concept:name`.

In a column-based CSV file, each row represents a trace, and the columns contain timestamps, indicating when the activity belonging to that column was executed. This structure of data is often encountered in healthcare settings, where standard forms that are used to log treatment steps use this structure. The plug-in `CSV to XES - trace per row` converts such a file into an XES event log, where each row becomes a trace, and every cell that has a timestamp is converted to an event (with the `concept:name` being the name of the column); every cell that does not parse as a timestamp becomes a trace attribute. This plug-in optionally takes a list of Java-based timestamp formats, such as `yyyy-M-d H:mm:ss.SSS`. Both of these CSV-plug-ins set some default log attributes, and attempt to guess the data type of each cell as accurately as possible.

Using the `map events` and `map traces` plug-ins, a particular event/trace attribute can be transformed using a provided map (in a separate CSV file), and written as another event/trace attribute.

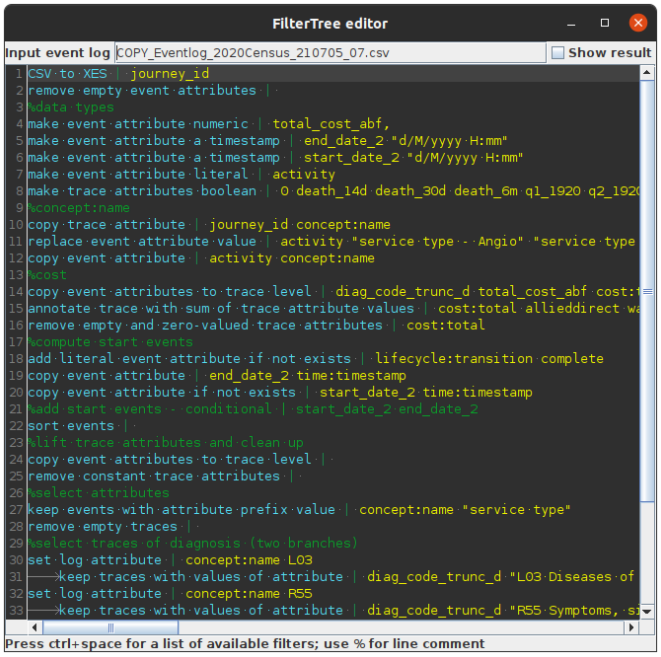


Fig. 1. Screenshot of the FilterTree editor interface.

The `add start events` filter copies each event, copies a chosen trace timestamp attribute to `time:timestamp` of the new event, sets `lifecycle:transition` to `start`, and adds a corresponding `concept:instance` to both events. The plug-in `sort events` sorts the events based on `time:timestamp`.

## III. Usage

### A. File Format

A filter tree is represented in a simple text file (with the .ftree extension). Comment lines start with %. The first line in this file contains the import event log, and each line thereafter contains one filter. On such a line, the name of the filter comes first, followed by the bar | symbol, followed by the parameters necessary for the filter, separated by spaces. If a parameter contains a space, it must be enclosed in double quotes. Indenting a filter line starts a new branch.

### B. User Interface

The user interface shows a filter tree file, with syntax highlighting and auto-completion; a screenshot is shown in Figure 1. When the user changes something by typing, after a small timeout the filters are automatically (re-)computed as necessary. The resulting logs, including all intermediate steps, are kept in a managed sub-folder; NB: the tool removes irrelevant files from this sub-folder. In the sub-folder, the final results of all branches are kept with a consistent and predictable filename, for compatibility with file management systems.

The user can enable a visualisation of the last log of the last branch, however, please note that this will attempt to load the log in memory; henceforth, this option is not enabled by default.

### C. Download

FilterTree is platform-independent (Java) and available with a GPL license from https://leemans.ch/filtertree. A full list of supported filters is included on this website, as well as a screencast demoing the tool. The source code is available at https://svn.win.tue.nl/repos/prom/Packages/SanderLeemans/FilterTree/. An empty text file can be used to start the editor.

## IV. Maturity

The FilterTree tool has been used in several of our own projects, including on private healthcare data (Figure 1), historical bureaucrat career data, road traffic fine collection data [2], etc. These settings ranged from simple (e.g. lifting a few event attributes to trace level) to complex (see Figure 1). In this latter case, the initial log was too complex (260MB CSV) to be of use directly in ProM or any other process mining tool we were allowed to try, due data confidentiality and semi-commercial nature of the data. The FilterTree tool allowed us to transform the log from CSV into XES and to filter it down to a manageable sub-view, which could be analysed in standard process mining tools.

## V. Conclusion

Preparing event data for analysis remains a rather ill-supported task, especially in settings with repeated small changes, large and complex event logs, data quality issues, or changing analysis questions. In this paper, we proposed a tool to edit XES and CSV files by means of filters. The FilterTree tool is repeatable as it keeps a full filter chain specificiation; it supports branching in the chain to allow multiple chains to share the same initial filters.

## References

[1] Alessandro Berti, Sebastiaan J Van Zelst, and Wil van der Aalst. Process mining for python (pm4py): bridging the gap between process-and data science. *arXiv preprint arXiv:1905.06169*, 2019.

[2] Sander J. J. Leemans, Shiva Shabaninejad, Kanika Goel, Hassan Khosravi, Shazia W. Sadiq, and Moe Thandar Wynn. Identifying cohorts: Recommending drill-downs based on differences in behaviour for process mining. In Gillian Dobbie, Ulrich Frank, Gerti Kappel, Stephen W. Liddle, and Heinrich C. Mayr, editors, *Conceptual Modeling - 39th International Conference, ER 2020, Vienna, Austria, November 3-6, 2020, Proceedings*, volume 12400 of *Lecture Notes in Computer Science*, pages 92–102. Springer, 2020.

[3] Artem Polyvyanyy. Process query language. In Artem Polyvyanyy, editor, *Process Querying Methods*, pages 313–341. Springer, 2022.

[4] Boudewijn F. van Dongen, Ana Karla A. de Medeiros, H. M. W. Verbeek, A. J. M. M. Weijters, and Wil M. P. van der Aalst. The prom framework: A new era in process mining tool support. In *Applications and Theory of Petri Nets 2005, 26th International Conference, ICATPN 2005, Miami, USA, June 20-25, 2005, Proceedings*, volume 3536 of *Lecture Notes in Computer Science*, pages 444–454. Springer, 2005.

[5] Maikel L. van Eck, Xixi Lu, Sander J. J. Leemans, and Wil M. P. van der Aalst. PM ^2 : A process mining project methodology. In *Advanced Information Systems Engineering - 27th International Conference, CAiSE 2015, Stockholm, Sweden, June 8-12, 2015, Proceedings*, volume 9097 of *Lecture Notes in Computer Science*, pages 297–313. Springer, 2015.

[6] H. M. W. Verbeek, Joos C. A. M. Buijs, Boudewijn F. van Dongen, and Wil M. P. van der Aalst. Xes, xesame, and prom 6. In *Information Systems Evolution - CAiSE Forum 2010, Hammamet, Tunisia, June 7-9, 2010, Selected Extended Papers*, volume 72 of *Lecture Notes in Business Information Processing*, pages 60–75. Springer, 2010.